CLAIMS

What is claimed is:

1   1.   A method of providing a protected execution environment on a computer

2   comprising:

3   intercepting an input/output request for a file from an application;

4   determining if the application is authorized to modify the protected execution

5   environment;

6   creating a redirected input/output request to an alternate environment when the

7   application is not authorized to modify the protected execution environment and the file

8   is within the protected execution environment; and

9   submitting the redirected input/output request to a file system manager.

1   2.   The method of claim 1 further comprising:

2   allowing the redirected input/output request to continue when it is intercepted.

1   3.   The method of claim 1 further comprising:

2   creating the protected execution environment.

1   4.   The method of claim 1 wherein the protected execution environment comprises a

2   directory for each of the applications that is authorized to modify the protected execution

3   environment.

1   5.   The method of claim 1 further comprising:

2   categorizing each application installed on the computer as authorized or not

3   authorized to modify the protected execution environment.

1   6.   The method of claim 1 wherein the alternate environment comprises a directory

2   associated with an application that is not authorized to modify the protected execution

3   environment.

1   7.   The method of claim 1 wherein the redirected input/output request specifies a

2   directory in the alternate environment that corresponds to a directory in the protected

3   execution environment specified in the input/output request.

1   8.   The method of claim 1, wherein a parent-child relationship is maintained between

2   an application that invokes another application.

1   9.   The method of claim 1, wherein determining if the application is authorized to

2   modify the protected execution environment comprises:

3   designating the application as not authorized to modify the protected execution

4   environment if the application was invoked by another application that is not authorized

5   to modify the protected execution environment.

1   10.   The method of claim 1, further comprising:

2   creating a null entry in a mirror directory structure for an executable for each

3   application authorized to modify the protected execution environment,

4   wherein determining if the application is authorized to modify the protected execution

5   environment comprises:

6   querying the existence of the executable for the application in the mirror

7   directory structure.

8   11.   The method of claim 10, further comprising:

1   maintaining an association between an executing application and a directory path

2   for the executable for the executing application,

3   wherein querying for the existence of the executable in the mirror data structure

4   comprises:

5   specifying the directory path for the executable associated with the

6   executing application.

1   12.   A method for operating a computer system with a protected execution

2   environment comprising:

3   executing a configuration utility to categorize a plurality of applications installed

4   on the computer system as authorized or not authorized to modify the protected execution

5   environment;

6   defining the protected execution based on the authorized applications; and

7   installing a protected execution agent in a file system to intercept input/output

8   requests submitted by the applications, wherein the protected execution agent directs an

9   input/output request to an alternate environment if the application that submitted the

10   request is not authorized and the request is directed to the protected execution

11   environment.

1   13.   The method of claim 12 wherein the configuration utility defines the protected

2   execution environment when categorizing the plurality of applications.

1   14.   The method of claim 12 wherein the alternate environment is defined based on at

2   least one application that is not authorized.

1 15. The method of claim 12, wherein the alternate environment is defined by the

2 configuration utility when categorizing the plurality of applications.

1 16. The method of claim 12, wherein the configuration utility further creates a null

2 entry in a mirror directory structure for an executable for each authorized application and

3 the protected execution agent further queries the existence of the executable for an

4 executing application in the mirror directory structure to determine if the application is

5 authorized.

1 17. The method of claim 16, wherein the protected execution agent further maintains

2 an association between the executing application and a directory path for the executable

3 for the executing application.

1 18. The method of claim 12, wherein the protected execution agent designates a

2 second application as not authorized if it was invoked by a first application that is not

3 authorized.

1 19. The method of claim 18, wherein the protected execution agent maintains a

2 parent-child relationship between the first and second applications.

1 20. The method of claim 12, wherein the protected execution agent is installed in a

2 hook chain in a file system manager to intercept the input/output requests before the

3 requests are processed by any other agent installed in the hook chain.

1   21.   The method of claim 12, wherein the configuration utility is executed prior to

2   providing the computer system to a user and the protected execution agent is installed

3   each time the computer system is booted.

1   22.   The method of claim 12, further comprising:

2   saving a copy of the protected execution environment; and

3   recovering from a failure of the computer system by replacing the protected

4   execution environment with the copy.

1   23.   The method of claim 22, wherein the copy is saved on the computer system in a

2   secure location.

1   24.   The method of claim 22, wherein the copy is saved on a remote computer server

2   and downloaded to the computer system.

1   25.   A method of determining a category for an application on a computer comprising:

2   categorizing the application as a first type;

3   creating a directory in a second directory structure for the application when it is a

4   first type, wherein the second directory structure mirrors a first directory structure that

5   contains an executable for the application;

6   creating a null entry for the executable for the application in the directory in the

7   second directory structure when the application is the first type;

8   querying the existence of the executable for the application in the second directory

9   structure, wherein the application is determined to be the first type when the executable

10   exists.

1   26.   A computer-readable medium having stored thereon computer-executable

2   instructions for performing a method comprising:

1   intercepting an input/output request for a file from an application;

2   determining if the application is authorized to modify the protected execution

3   environment;

4   creating a redirected input/output request to an alternate environment when the

5   application is not authorized to modify the protected execution environment and the file

6   is within the protected execution environment; and

7   submitting the redirected input/output request to a file system manager.

1   27.   The computer-readable medium of claim 26 having further computer-readable

2   instructions comprising:

3   allowing the redirected input/output request to continue when it is intercepted.

1   28.   The computer-readable medium of claim 26 having further computer-readable

2   instructions comprising:

3   categorizing each application installed on the computer as authorized or not

4   authorized to modify the protected execution environment.

1   29.   The computer-readable medium of claim 26 having further computer-readable

2   instructions comprising:

3   creating the protected execution environment from a directory for each of the

4   applications that is authorized to modify the protected execution environment.

1   30.   The computer-readable medium of claim 26 having further computer-readable

2   instructions comprising:

3   creating the alternate environment from a directory associated with an application

4   that is not authorized to modify the protected execution environment.

1   31.   The computer-readable medium of claim 26 having further computer-readable

2   instructions comprising:

3   storing a directory path specified in the input/output request in the redirected

4   input/output request to direct the request to a corresponding directory path in the alternate

5   environment.

1   32.   The computer-readable medium of claim 26 having further computer-readable

2   instructions comprising:

3   maintaining a parent-child data structure to track between relationships between

4   applications that invoke other applications.

1   33.   The computer-readable medium of claim 26 having further computer-readable

2   instructions comprising:

3   designating the application as not authorized to modify the protected execution

4   environment if the application was invoked by another application that is not authorized

5   to modify the protected execution environment.

1   34.   The computer-readable medium of claim 26 having further computer-readable

2   instructions comprising:

3   creating a null entry in a mirror directory structure for an executable for each

4   application authorized to modify the protected execution environment; and

1   querying the existence of the executable for the application in the mirror directory

2   structure when determining if the application is authorized to modify the protected

3   execution environment

1   35.   The computer-readable medium of claim 34 having further computer-readable

2   instructions comprising:

3   maintaining an association between an executing application and a directory path

4   for the executable for the executing application; and

5   specifying the directory path for the executable associated with the executing

6   application when querying for the existence of the executable in the mirror data structure.

1   36.   A computer system comprising:

2   a processing unit;

3   a memory coupled to the processing unit through a system bus;

4   a computer-readable medium coupled to the processing through the system bus;

5   and

6   a protected environment agent executing from the computer-readable medium,

7   wherein the protected environment agent causes the processing unit to intercept

8   input/output requests submitted by applications executing on the computer system and

9   further causes the processing unit to redirect each input/output request to an alternate

10   environment if the application that submitted the request is not authorized to modify a

11   protected execution environment and the request is directed to the protected execution

12   environment.

1   37.   The computer system of claim 36 further comprising:

2   a configuration utility executing from the computer-readable medium, wherein the

3   configuration utility causes the processing unit to categorize each application installed on

4   the computer system as authorized or not authorized to modify the protected execution

5   environment and further to cause the processing unit to define the protected execution

6   environment to contain directories associated with the authorized applications.